A comprehensive review on ensemble deep learning: Opportunities and challenges

Ammar Mohammed, Rania Kora

Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt

A B S T R A C T

In machine learning, two approaches outperform traditional algorithms: ensemble learning and deep learning. The former refers to methods that integrate multiple base models in the same framework to obtain a stronger model that outperforms them. The success of an ensemble method depends on several factors, including how the baseline models are trained and how they are combined. In the literature, there are common approaches to building an ensemble model successfully applied in several domains. On the other hand, deep learning-based models have improved the predictive accuracy of machine learning across a wide range of domains. Despite the diversity of deep learning architectures and their ability to deal with complex problems and the ability to extract features automatically, the main challenge in deep learning is that it requires a lot of expertise and experience to tune the optimal hyperparameters, which makes it a tedious and time-consuming task. Numerous recent research efforts have been made to approach ensemble learning to deep learning to overcome this challenge. Most of these efforts focus on simple ensemble methods that have some limitations. Hence, this review paper provides comprehensive reviews of the various strategies for ensemble learning, especially in the case of deep learning. Also, it explains in detail the various features or factors that influence the success of ensemble methods. In addition, it presents and accurately categorized several research efforts that used ensemble learning in a wide range of domains.

© 2023 Published by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

1. Introduction ........................................................................................................... 758
2. Trends of ensemble learning .................................................................................. 758
3. Foundations of ensemble learning ........................................................................ 760
   3.1. Data sampling ............................................................................................. 761
   3.2. Training baseline classifiers ........................................................................ 762
   3.3. Fusion method ......................................................................................... 762
       3.3.1. Voting method .................................................................................. 763
       3.3.2. Meta learning method ...................................................................... 763
4. Ensemble methods ................................................................................................ 764
   4.1. Common ensemble methods ....................................................................... 764
       4.1.1. Bagging ........................................................................................... 764
       4.1.2. Boosting ......................................................................................... 764
       4.1.3. Stacking .......................................................................................... 764

Peer review under responsibility of King Saud University.

E-mail addresses: ammar@cu.edu.eg (A. Mohammed), rania.kora@pg.cu.edu.eg (R. Kora)

https://doi.org/10.1016/j.jksuci.2023.01.014
1319-1578/© 2023 Published by Elsevier B.V. on behalf of King Saud University.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
1. Introduction

In a world full of diverse and varied data sources, Machine learning has become one of the most important and dominant branches of artificial intelligence methods, which is applied in many fields. There are many different learning algorithms and methods. Each method's pitfalls and drawbacks are measured in terms of several factors, including performance and scalability. Based on a lot of research in machine learning, two methods dominate learning algorithms; namely deep learning (Deng et al., 2014) and ensemble learning (Polikar, 2012; Sagi and Rokach, 2018; Rokach, 2019). The deep learning techniques can scale and handle complex problems and offer an automatic feature extraction from unstructured data (Kamilaris and Prenafeta-Boldú, 2018). Also, deep learning methods contain several types of network architectures for different tasks, such as feeding forward neural networks (Bebis and Georgiopoulos, 1994), convolutional neural networks (Collobert and Weston, 2008), recurrent neural networks (Yu et al., 2019). Many others (Ain et al., 2017). However, the training process of deep learning models requires a massive effort, and tuning the optimal hyper-parameters requires expertise and extensive trial, which is a tedious and time-consuming task. Also, training more complex deep neural network increases the chance of overfitting.

Ensemble Learning, on the other hand, refers to a learning methodology that combines several baseline models to build a bigger single yet more powerful model than its constituents (Kumar et al., 2021). Also, ensemble learning can reduce the risk of overfitting thanks to the diversity of baseline models. Ensemble learning was successfully applied in various fields and domains and outperforms single models (Anwar et al., 2014; Shahzad and Lavesson, 2013; Prusa et al., 2015; Ekbal and Saha, 2011). There are several ensemble techniques varied in terms of how different baseline models are trained and combined. The most widely used ensemble techniques include averaging, bagging, random forest, stacking, and boosting. In the literature, there are many reviews about ensemble learning methods, and techniques (Krawczyk et al., 2017; Sagi and Rokach, 2018; Dong et al., 2020). Traditional ensemble learning is based on integrating traditional machine learning models and applying them in different fields (Tsai et al., 2011; Abellán and Mantas, 2014; Catal et al., 2015; Da Silva et al., 2014; Aburoumman and Reaz, 2016). However, these efforts were limited to simple single models. In recent years, numerous attempts have been made to approach ensemble learning to deep learning (Haralabopoulos et al., 2020; Tasci et al., 2021; Alharbi et al., 2021; Ortiz et al., 2016; Can Malli et al., 2016; Xu et al., 2016). However, most of these attempts are articulated using the average voting method of baseline deep learning models. However, the ensemble process using average voting methods is biased towards weak baseline learners and is not a smart strategy for combing the baseline learners. Despite several strategies of combining baseline learners that can be applied to ensemble deep learning, these strategies have some limitations in terms of generalization, difficulties in training, and other issues (Tasci et al., 2021).

In the literature, some review efforts have introduced the concept of deep ensemble learning (Dong et al., 2020; Sagi and Rokach, 2018). This effort, however, is restricted to the application of ensemble in particular domains with reviews on traditional ensemble approaches.

To this end, this paper tries to comprehensively review the different strategies for applying ensemble deep learning. It also presents several aspects that influence the success of ensemble methods, such as the type of utilized baseline learning models, the data samples techniques used in training, the diversity of employing different baseline classifiers, and the fusion methods of the baseline deep models. Additionally, it discusses the benefits and drawbacks are each strategy.

The contributions of this paper are highlighted as the following. First, we provide quantitative analytics insight into ensemble learning. Second, we introduce the fundamental concepts and general architecture of ensemble learning, strategies for generating diversity among the base classifiers, and the factors impacting any ensemble method. Additionally, we present the structure of each of the several ensemble methods and the advantages, disadvantages, and general classifications for each method. Moreover, we discuss the different strategies of ensemble deep learning models. Finally, we comprehensively survey numerous research efforts that used ensemble learning in various applications.

The remainder of this manuscript is organized as the following: Section 2 introduces quantitative analytics for research discussions on ensemble learning and deep learning techniques indexed in “Scopus.” Section 3 introduces a comprehensive overview of the foundations of ensemble learning and the factors that influence any ensemble method. Section 4 provides an overview of various methods in ensemble learning and explains the general strategies of ensemble based on deep learning models. Section 5 discusses several criteria for evaluating different ensemble learning methods. Section 6 reviews several applications of ensemble learning in different domains. Finally, Section 7 concludes this paper and gives directions for future trends.

2. Trends of ensemble learning

Due to the strength and effectiveness of the ensemble learning system to improve the predictive performance of models. Ensemble learning has become an important research trend in recent years, which has led to an increase in the number of research used for ensemble learning in several domains of applications. Hence, this section presents this important trend in one of the most powerful databases, “Scopus.” To show the extent to which the articles indexed published for ensemble learning increased each year and the different applied fields of ensemble learning from 2014 to 2021. The search query in this database is “Ensemble Learning” and “Ensemble Deep Learning.” These were searched in the article
titles, abstract, and keywords. Fig. 1 shows the number of articles published for the search term “Ensemble Learning” each year in the abovementioned period. The figure shows that the number of articles found using this term was estimated at 25,262, indicating an increase in the ensemble learning trend over several years. In addition, Fig. 2 shows the number of articles that discussed the search term “Ensemble Learning” in all fields. From the figure, it can be noted that the field of computer sciences has the highest estimated number of articles mentioned, estimated as 16,782 documents. Fig. 3 shows the number of articles published for the search term “Ensemble Deep Learning” each year in the abovementioned period. The figure shows that the number of articles found using this term was estimated as 6,173, indicating increased interest from researchers in this trend. Also, Fig. 4 shows the number of articles that discussed the search term “Ensemble Deep Learning in all fields. From the figure, it can be noted that the field of computer
sciences has the highest estimated number of articles mentioned, estimated at 4520 documents.

According to the above statistical information, it is clear that research in ensemble learning and ensemble deep learning is growing faster each year due to its ability to improve prediction performance. According to estimates, the largest number of articles using “Ensemble Learning” and “Ensemble Deep Learning” in 2021 was estimated at 7160 and 2340 documents, respectively. In addition, ensemble learning and deep ensemble learning have been applied in several fields, especially computer science, with the highest utilization rate of ensemble learning and deep ensemble learning of 30% and 35.1%, respectively.

3. Foundations of ensemble learning

The general framework of any ensemble learning system is to use an aggregation function $G$ to combine a set $h$ of baseline classifiers, $c_1, c_2, \ldots, c_h$, towards predicting a single output. Given a dataset of size $n$ and features of dimension
\[ m. D = \{ (x_i, y_i) \}_{1 \leq i \leq n}, x_i \in \mathbb{R}^m \text{, the predication of the output} \]
\[ y_i = \phi(x_i) = G(c_1, c_2, \ldots, c_i) \quad (1) \]

Fig. 5 illustrates the general abstract framework of ensemble learning. All ensembles are made up of a collection of baseline classifiers (classifiers ensemble) that have been trained on input data that produce predictions that are combined to produce an aggregate prediction (Lakshminarayanan et al., 2017). Ensemble strategies differ on how to select the baseline classifiers that are trained. Two strategies generate diversity among the base classifiers based on their nature, either homogeneous or heterogeneous ensembles as shown in Fig. 6 (Seijo-Pardo et al., 2017). Homogeneous ensemble (da Conceiçao et al., 2015) consists of baseline classifiers of the same type, with each classifier based on different data. The feature selection method in this strategy is the same for different training data. The main difficulty in homogeneous form is the generation of diversity from the same learning algorithm. Whereas heterogeneous ensembles consist of different numbers of baseline classifiers, (da Conceiçao et al., 2016), as each classifier is based on the same data. In heterogeneous classifiers, the feature selection method is different for the same training data. Finally, homogeneous ensemble methods are more appealing to researchers since they are easier to understand and apply. Also, it is less costly to build homogeneous ensembles than heterogeneous ones (Hosni et al., 2019).

Generally, any ensemble framework can be viewed and defined using three characteristics that affect its performance. The first one is the dependency on the trained baseline models, whether they are sequential or parallel. The second characteristic is the fusion methods, which involve choosing a suitable process for combining outputs of the baseline classifiers using different weight voting or meta-learning method. The third characteristic is the heterogeneity of the involved baseline classifiers, whether homogeneous or heterogeneous. Table 1 summarizes the characteristics of the popular ensemble methods. In what follows, those characteristics will be discussed in detail.

### 3.1. Data sampling

The selection of a data sampling method is one of the most important factors affecting the performance of the ensemble system. In the ensemble system, we need diversity in the data sampling decisions of the baseline classifiers. There are two strategies of the sampling methods from the training dataset in the ensemble system: the independent datasets strategy and the

<table>
<thead>
<tr>
<th>Method</th>
<th>Dependent</th>
<th>Fusion method</th>
<th>Heterogeneity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagging</td>
<td>Parallel</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>Random Forest</td>
<td>Parallel</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>Boosting</td>
<td>Sequential</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>Sequential</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>Sequential</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>Extreme Gradient</td>
<td>Sequential</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>Boosting</td>
<td>Sequential</td>
<td>Weight Voting</td>
<td>Homogenous</td>
</tr>
<tr>
<td>Stacking</td>
<td>Parallel</td>
<td>Meta Learning</td>
<td>Heterogeneous</td>
</tr>
<tr>
<td>Hybrid Ensemble</td>
<td>Both</td>
<td>Both</td>
<td>Heter/Homogeneous</td>
</tr>
</tbody>
</table>

![Fig. 5. General Framework of Ensemble.](image)

![Fig. 6. General framework of homogeneous and heterogeneous ensemble.](image)
dependent datasets strategy (Sagi and Rokach, 2018). In independent datasets strategy, (Ge et al., 2020), are those subsets that are not dependent on each other. By contrast, independent datasets strategy (Hassan et al., 2013) are subsets dependent on each other. The main advantage of using an independent datasets strategy is that its sub-data set is not affected by the performance of other sub-datasets, in contrast to using a dependent datasets strategy, where its sub-data set is affected by the results of the previous sub-data set. The difficulty of the data sampling method in both strategies is determining the optimal size of each data sample and the maximum number of samples. In addition, determining the appropriate strategy for data samples according to different ensemble methods (Lu and Van Roy, 2017).

3.2. Training baseline classifiers

The diversity of the baseline classifiers is the second influential factor in the ensemble system. At the core of any ensemble-based system are two techniques for training individual ensemble members: the sequential ensemble technique and the parallel ensemble technique (Huang et al., 2016). In sequential ensemble technique (Sultana et al., 2020), different learners learn sequentially because of data dependency. Thus, the errors made by the first model are sequentially corrected by the second model as shown in Fig. 7. So, the main advantage of sequential methods is to exploit the dependence between the base learners (Saeed et al., 2022). Whereas in parallel ensemble technique (Tang et al., 2020), base learners are generated simultaneously, as there is no data dependency. So, each data in the base learner is generated independently as shown in Fig. 8. This technique’s basic advantage is exploiting the independence between base learners. Thus, the errors made by one model differ from those found in another independent model, allowing the ensemble model to calculate the average out the errors (Valle et al., 2010).

3.3. Fusion method

Output fusion refers to integrating the outputs of the baseline classifiers into a single output. There are two methods of fusion, the voting method, and the meta-learning method. We will explain in each method how to implement in integrating the outputs of baseline classifiers, their advantages, and the difficulty of applying them, as well as select the appropriate fusion method for each of the ensemble methods. The fusion methods can be used with independent or dependent data samples and can also be used with parallel or sequential baseline classifiers.

3.3.1. Voting method

Voting methods are generally used in classification or regression problems to improve predictive performance. In addition, voting methods are the appropriate integrating method for bagging and boosting methods. The first fusion method is a voting ensemble, which includes three methods: max voting, averaging voting, and weighted average voting. We will discuss in each voting method the nature of implementation and the advantages and drawbacks of implementing it.

1. Max Voting: The first and most popular voting method is the max voting (Kim et al., 2003) often, often known as majority voting or hard voting. The idea of max voting involves collecting predictions for each class label and predicting the class label with the most votes as shown in function (2). For example,
assuming we combine three classifiers, C1, C2, and C3, that assign the following classifications to a training sample: [0.0, 1] becomes y = mode [0.0, 1] = 0. We would categorize the sample as “class 0”. Max voting is often used in the bagging method. Another type of max voting is soft voting. Soft voting involves collecting predicted probabilities for each class label and predicting the class label with the largest probability as shown in function (3). Max voting is distinguished from soft voting in that once we know the prediction for any of the baseline classifiers, we do not need to store any other information about the probability distributions of the predictions. On the other hand, soft voting needs to store and use all the distribution values, making it more computationally and costly for storage. However, in soft voting, we can use various methods to calculate the prediction, such as calculating maximum or average probability values (Delgado, 2022). In general, the max voting method has the advantages of being simple to understand and the simplest method of voting. The drawbacks of the max voting method include the computational expense of using several baseline models. Additionally, max voting is useless when the baseline classifiers predictions are the same results and may not fit all problems (Nti et al., 2020).

\[
y^* = \text{mode}(C_1(x), C_2(x), ..., C_n(x)) \quad (2)
\]

Where \( y^* \) a predict the class label via majority (plurality) voting of each classifier \( C_n \).

\[
y^* = \arg \max_{i} \sum_{j=1}^{n} w_j P_{ij} \quad (3)
\]

Where \( w_j \) is the weight that can be assigned to the \( j^{th} \) classifier.

2. **Averaging Voting**: The second voting method is the averaging voting (Montgomery et al., 2012). The idea of averaging voting is that predictions are extracted from multiple models, and an average of the predictions is used to make the final prediction. Average prediction is calculated using the arithmetic mean, which is the sum of the predictions divided by the total predictions made as shown in function (4). For instance, suppose the ensemble of classifiers contained three members: C1(x)= [0.9, 0.1], C2(x)= [0.2, 0.8], and C3(x)= [0.6, 0.4]. The mean prediction would be as follows: to calculate the class 0 \( y^0 = (0.9 + 0.2 + 0.6)/3 \) = 0.566. And to calculate the class 1 \( y^1 = (0.1 + 0.8 + 0.4)/3 \) = 0.433, would yield a prediction \( y^* = 0 \). The average voting method has the advantage of being the strongest from the point of view of predictive power. In addition, it is more accurate in performance than majority voting and reduces overfitting. Also, the average voting is a natural competitor to the max voting for bagging method. The drawbacks of the average voting method include being computationally more expensive than the max voting method, as it requires averaging the prediction results of all the baseline models. One limitation of the averaging voting method is that it assumes that all baseline models in the ensemble are equally effective. However, it is not the case as some models may be better than others (Hopkinson et al., 2020).

\[
y^* = \arg \max_{i} \frac{1}{n} \sum_{j=1}^{n} w_{ij} \quad (4)
\]

where \( w_{ij} \) is the probability of the \( i^{th} \) class label of the \( j^{th} \) classifier.

3. **A weighted Average Voting**: The third method of voting is the weighted average voting, which is a slightly modified version of averaging voting (Latif-Shabgahi, 2004). The idea of weighted average voting is different weights given to the baseline learners, indicating the importance of each model in prediction. By multiplying each prediction by the weight of the classifiers to produce a weighted sum and then dividing the result by the sum of the weights of the classifier, these weights may be used to calculate the weighted average for each class 0 or class 1 as shown in function (5). For instance, suppose the ensemble of classifiers contained three members: C1(x)= [97.2, 2.8], C2(x)= [100.0, 0], and C3(x)= [95.8, 4.2]. It has constant weights for ensemble members [0.84, 0.87, 0.75]. To calculate the class 0 \( y_0^* = ((97.2 * 0.84) + (100.0 * 0.87) + (95.8 * 0.75)) / (0.84 + 0.87 + 0.75) \) = 97.763. And to calculate the class 1 \( y_1^* = ((2.8 * 0.84) + (0 * 0.87) + (4.2 * 0.75)) / (0.84 + 0.87 + 0.75) \) = 2.235, would yield a prediction \( y^* = 0 \). The weighted average voting method is more accurate than the simple average-voting method. The challenge in using a weighted average ensemble is choosing each member’s relative weighting. Also, the computation is more expensive than the average voting method, as it requires calculating the weighted average of the prediction results of all the baseline models, which makes it of little application (Khan et al., 2020).

\[
y^* = \frac{\sum_{j=1}^{m} w_j x_j}{\sum_{j=1}^{m} w_j} \quad (5)
\]

where \( w \) weighted average, \( m \) is a number of terms to be averaged, weights applied to x values \( w_j \), and data values to be averaged \( x_j \).

3.3.2. **Meta learning method**

The second fusion method is meta-learning (Soares et al., 2004), also known as “learning to learn”, which is the process of learning from learners. The term “meta-learning” covers learning based on previous experience with other tasks. Therefore, it is used to improve the performance and results of a learning algorithm by changing some aspects of the learning algorithm based on experiment results. The meta-learning method differs from traditional machine-learning models in that it involves more than one learning stage where the individual inducer outputs serve as an input to the meta-learner that generates the final output (Kuruvayil and Palaniswamy, 2021).

Over the past five years, interest in meta-learning has increased, especially after 2017. With the increased use of advanced machine learning algorithms, the difficulties of training these learning algorithms have led to an increased interest in meta-learning. Machine learning algorithms have many challenges, such as the high operational costs due to many experiments during the training phase, which takes a long time to find the best model that achieves the best performance for a certain dataset. Meta-learning helps to meet these challenges by improving learning algorithms and finding learning algorithms that perform better (Kuruvayil and Palaniswamy, 2022). In addition, the benefits of meta-learning include speeding up learning processes by reducing the number of experiments required, helping learning algorithms better adapt to changing conditions, and optimizing hyperparameters to achieve optimal results. Moreover, this method provides an opportunity to tackle many challenges of deep learning, including data size, computational complexities, and generalization. The challenge in meta-learning is to learn from experience in a systematic, data-driven manner (Hospedales et al., 2021). There are many meta-learning methods, the most common of which is stacking (Haghigi and Omranoosh, 2021). To implement the meta-learning, there are several challenges represented in defining an appropriate meta-learning approach and the computation time complexity, whether through a large amount of available dataset.
or through multiple baseline models or multiple levels of meta-
learning (Monteiro et al., 2021).

4. Ensemble methods

This section presents two aspects. The first aspect includes the
structure of the most popular ensemble learning methods and lists
each method’s benefits, drawbacks, and implementation chal-
 lenges separately. The second aspect presents the idea of deep
ensemble learning and the advantages of its application compared
to traditional ensemble learning. It also discusses the deep learning
challenges that ensemble deep learning overcomes them. More-
over, it introduces the different strategies for applying ensemble
deep learning and the advantages of each strategy with an expla-
nation of the factors that can affect its performance.

4.1. Common ensemble methods

Three popular ensemble learning methods can be used to
improve the machine learning process: bagging, boosting, and
stacking. We will discuss the nature of each method’s work and
its characteristics regarding the nature of data generation, the
nature of training of baseline classifiers, and the appropriate fusion
methods. In addition, the benefits, drawbacks, and implementation
challenges of each method will be covered.

4.1.1. Bagging

The bagging method (Breiman, 1996), also known as bootstrap
aggregating, is a completely data-specific algorithm. It refers to
creating multiple small subsets of data from the actual dataset.
The goal of bagging is to create more diverse predictive models
by adjusting a stochastic distribution of the training datasets,
where small changes in the training data set will lead to significant
changes in the model predictions. Bagging is shorthand for the
combination of bootstrapping and aggregating. In bootstrapping,
the training of the ensemble models on bootstrap replicates the
training dataset. In aggregation, the final result is achieved by
majority voting of the model’s predictions performed to determine
the final prediction. Bagging offers the advantage of reducing vari-
ce, thus eliminating overfitting. It also performs well on high-
dimensional data. The drawback of bagging is that it is computa-
tionally expensive and has high bias, and it also leads to a loss of
interpretability of a model (Bühlmann and Yu, 2002). Random For-
est (RF) algorithm (Breiman, 2001) is a good example of bagging.
There are several challenges to implementing the bagging method:
determining the optimal number of base learners and subsets and
the maximum number of bootstrap samples per subset. In addi-
tion, the determination of fusion method of integrating the outputs
of the base classifiers from various voting methods. In summary,
the bagging method uses parallel ensemble techniques where
baseline learners are generated simultaneously, as there is no data
dependency and the fusion methods depend on different voting
methods. The function of bagging is shown as follows (6):

\[ f(x) = \frac{1}{B} \sum_{B=1}^{B} f_{B}(x) \]  

where \( f_{B}(x) \) weak learners, \( \frac{1}{B} \) generates bootstrapping sets.

4.1.2. Boosting

Boosting method was first presented by Freund and Schapire in
the year 1997 (Freund et al., 1996), and is a sequential process
where each subsequent model attempts to correct the errors of
the previous model. Boosting consists of sequentially multiple
weak learners in a very adaptive way, whereby each model in
the sequence is fitted, giving more importance to observations in
the dataset that the previous models in the sequences badly han-
dled. Boosting, like bagging, can be used for regression and classi-
fication problems. Boost algorithms include three types, namely,
Adaptive Boosting (AdaBoost) (Freund et al., 2003), Stochastic Gra-
dient Boosting (SGB) (Friedman, 2001), and Extreme Gradient
Boosting (XGB), also known as XGBoost (Friedman et al., 2000).
Several studies have applied various types of boosting. For exam-
ple, the AdaBoost algorithm is implemented in Sun et al. (2016)
for noise detection and in Asbai and Amrouche (2017) for speech
feature extraction. The XGB algorithm is implemented in
Haumahu et al. (2021) for Fake news classification. The SGB algo-
rithm is implemented in Shin (2019) for early prediction of safety
accidents at construction sites. Boosting provides ease of interpre-
tation of the model and helps reduce variance and bias in a
machine learning ensemble. The drawback of boosting is that each
classifier must fix the errors in the predecessors. To implement
boosting, several challenges are represented by the difficulty of
scaling sequential training in boosting. It is computationally costly
and more vulnerable to overfitting when increasing the number of
iterations. Finally, it can be noted that boosting algorithms can be
slower to train when compared to bagging because a large number
of parameters can also affect the behavior of the model. In sum-
mary, the boosting method uses sequential ensemble techniques
where different learners learn sequentially, as there is data depen-
dency and the fusion methods depend on different voting methods.
The function of boosting is shown as follows (7):

\[ f(x) = \sum_{t} \alpha_{t} h_{t}(x) \]  

where creates a strong classifier \( f(x) \) from several weak classifiers
\( h_{t}(x) \). This is done by building a model from the training data, then
creating a second model that attempts to correct the errors from
the first model \( \alpha_{t} \).

4.1.3. Stacking

Stacking method (Smyth and Wolpert, 1997), also known as
Stacked Generalization, is a model ensembling technique used to
combine information from multiple predictive models to generate
a new model (meta-model). The architecture of a stacking model
involves two or more base models, referred to as a level-0 model,
and a meta-model that combines the predictions of the base mod-
els, referred to as a level-1 model. In level 0 models (base models),
models fit on the training data and whose predictions are com-
plied. However, in the level 1 model (meta-model), the model
learns how to combine the base models’ predictions best. The out-
puts from the base models used as input to the meta-model may be
probability values, or class labels in the case of classification (Ma
et al., 2018). The stacking method typically performs better than
all trained models. For instance, a stacking ensemble learning sys-
tem was proposed by Divina et al. (2018) to forecast electric energy
usage in Spain and Qiu et al. (2014) to forecast electric energy
usage in Australia. Stacking has the benefit of a deeper comprehen-
sion of the data, making it more precise and effective. Overfitting is
a major issue with model stacking because there are so many pre-
dictors that all predict the same target that is merged. In addition,
multi-level stacking is costly to data (as lots of data needed to be
trained) and time-consuming (as each layer adds multiple models)
(Xiong et al., 2021). To implement stacking, several challenges are represented by identifying the appropriate
number of baseline models and the baseline models that can be
relied upon to generate better predictions from datasets when
designing a stacking ensemble from scratch. Also, the difficulty of
interpreting the final model and the computation time complexity
are added when the amount of available data grows exponentially.
A highly complex model would take months to run. Finally, the
problem of multi-label classification raises many issues, such as
overfitting and the curse of dimensionality, from the high dimensionality of the data (Chatzimparmpas et al., 2020). In summary, the stacking method uses parallel ensemble techniques where baseline learners are generated simultaneously, as there is no data dependency and the fusion methods depend on the meta-learning method. The function of stacking is shown as follows (8):

\[ f_s(x) = \sum_{i=1}^{n} a_i f_i(x) \]  

(8)

A formal stacking concept: Here, we make predictions from several models \((m_1, m_2, m_3, \ldots, m_n)\) to build a new model, where the new model is used to make predictions on the test dataset. Stacking seeks to increase the predictive power of a model. The basic idea of stacking is to “stack” the predictions of \((m_1, m_2, m_3, \ldots, m_n)\) by a linear combination of weights \(a_1, \ldots, (i = 1, 2, \ldots, n)\).

4.2. Ensemble deep learning

In recent years, deep learning or deep neural learning has led to a series of achievements in various tasks (Arel et al., 2010). Deep learning architectures have shown great success in almost all challenges related to machine learning across different areas, such as NLP (Mohammed and Kora, 2019; Elnagar et al., 2020), computer vision (Haque et al., 2020; Brunetti et al., 2018), speech recognition (Jaouedi et al., 2020; Noda et al., 2015), Machine translation (Popel et al., 2020; Popel et al., 2020). Deep neural network models are nonlinear methods that learn through a stochastic training algorithm. This means that it is highly flexible, able to learn the complex relationships between variables and approximate any mapping function. The downside to this flexibility is that the models need a higher variance. The high variance of the deep model can be addressed by ensemble deep learning approach opportunities by training multiple deep models for the problem and combining their predictions. Hence, ensemble deep learning methods refer to training several baseline deep models and combining some rules to make predictions. Ensemble deep learning aims to effectively combine the major benefits of several deep learning models with those of an ensemble learning system (Mohammed and Kora, 2021). Despite the power of ensemble deep learning system methods in improving prediction performance, most of the ensemble deep learning literature focuses on only applying a majority of voting algorithms to enhance the performance due to its simplicity.

Ensemble learning based on deep learning models is more difficult than ensemble learning based on traditional classifiers due to deep neural networks containing millions to billions of hyper-parameters that need a lot of time and space to train multiple base deep learners. Thus, hyper-parameters are challenges in the application of ensemble deep learning techniques. Ensemble learning strategies are formed in the context of manipulating the data level or the baseline model level. In manipulation at the level of data, by sampling data or cross-validation data (re-sampling) to create new training sets to train different base learners. In manipulation at the level of basic models, deep learning is distinguished by more diverse strategies than traditional or machine learning, which is the possibility of reducing the number of hyper-parameters used in the ensemble base deep models by selecting the same model and changing the hyper-parameters (Saleh et al., 2022). Fig. 9 shows four strategies through which deep learning can be conducted based on the ensemble represented by: (A) Applying many different basic models using the same data. (B) Applying different structures of the same basic model using the same data. (C) Applying many different basic models using many different data samples. (D) Applying different structures of the same basic model using many different data samples. Comparing these strategies shows that strategy A and strategy C are compatible with deep learning models and traditional learning techniques. Whereas strategy B and strategy D only apply to deep learning models and cannot be used with traditional learning techniques, making the ensemble deep learning strategies diverse. In addition, strategy B and strategy D enable ensemble deep learning to reduce the hyper-parameters of the baseline deep models by different structures of the same basic model by altering some of the hyper-parameters values. In addition to these strategies, the strength of the ensemble deep learning system depends on the ensemble system design, from identifying the most effective deep learning models to address the problem and determining the appropriate

Fig. 9. Different cases of ensemble deep learning.
number of baseline deep learning models, such as three or more and also determining the optimal ratio for data splitting such as (80–20 or 70–30 or 60–40). Moreover, we consider factors that may affect the deep ensemble system, such as defining the nature of data generation, training deep baseline models, and deciding the most appropriate fusion method of combining outputs of the baseline classifiers, as previously mentioned. These three factors affect the general framework of the ensemble system.

5. Evaluating ensembles

With the emergence of ensemble learning approaches, lots of research has been conducted to evaluate the methods of ensemble (Hashino et al., 2007; Zhang et al., 2016; Das and Sengur, 2010; Hosni et al., 2019). The evaluation is crucial to determining the effectiveness of a certain ensemble method. There are several criteria for evaluation ensemble, including predictive performance. Other criteria, such as the computational complexity or the comprehensibility of the generated ensemble, can also be important. In the following, we summarize the different evaluation criteria of ensemble learning.

5.1. Predictive performance

Predictive performance metrics have always been the primary criterion for choosing the performance of classifiers. Also, predictive performance measures are considered objective and quantifiable, so they are often used to benchmark machine learning algorithms practically. The first step to applying predictive performance is to use a suitable dataset. The holdout technique is a typical approach for measuring predictive performance where the given dataset is randomly divided into two subsets: training and test sets. Other versions of the holdout method might be utilized. It is normal procedure to resample data, which means dividing it into training and test sets in different ways. Two common resampling methods include random subsampling, and n-fold cross-validation (Dai, 2013).

There are common measures for evaluating an ensemble model. Accuracy is one of the popular and simplest metrics, as defined in Eq. 9:

\[
\text{Accuracy} = \frac{\text{number of true predictions}}{\text{total number of prediction}}
\] (9)

In some cases, accuracy is insufficient and can be deceptive in evaluating an ensemble model with imbalanced class distributions. In the latter scenario, other measures can be used as alternative measures, such as Recall, Precision, Specificity, and F-Measure (Kadam et al., 2019).

Recall, also known as sensitivity, measures the ensemble model’s capability to identify positive samples, which as defined in Eq. 10:

\[
\text{Recall} = \frac{\text{true positive}}{\text{positive}}
\] (10)

where true positive denotes the number of true positive observations and positive denotes the number of positive observations.

Another well-known performance metric is precision. It quantifies how many instances classified as positive are actually positive. Formally, the precision equation is defined as 11:

\[
\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}
\] (11)

Likewise, specificity measures how well the model identifies negative samples. The equation is defined as 12:

\[
\text{Specificity} = \frac{\text{true negative}}{\text{negative}}
\] (12)

where true negative denotes the number of true negative observations and negative denotes the number of negative observations.

There is commonly a trade-off between precision and recall metrics. Attempting to enhance one measure often results in the fall of the second. Thus, F-Measure quantifies this trade-off by calculating the harmonic mean of both precision and recall. More specifically, this measure is defined in Eq. 13:

\[
F - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\] (13)

5.2. Computational complexity

The computational complexity of the ensemble approach is an additional essential aspect to consider. Generally, the computational cost refers to the amount of CPU time required by each ensemble model. The computational cost is distributed on two complexity metrics: The computational cost of training and creating the ensemble model and the computational cost of predicting a new instance: The computational cost of the prediction is relatively small compared to the computational cost of the training ensemble. Thus, this metric should be addressed. In terms of memory, a smaller ensemble model needs less memory to keep its components. Furthermore, smaller ensembles perform faster prediction.

5.3. Other criteria

In addition to computational complexity and prediction accuracy, other considerations may be made when selecting the best ensemble method. These criteria include Interpretability, Scalability, usability, and robustness of the ensemble model. Interpretability (Carvalho et al., 2019) refers to the ability of a user to understand the ensemble outcomes. However, interpretability is typically a subjective metric. One of the many quantitative metrics and indicators that can help us evaluate this criterion is the compactness metric. Compactness in the ensemble can be evaluated using the number of classifiers involved and the complexity of each classifier.

On the other hand, scalability refers to the capacity of the ensemble approach to construct a classification model given large amounts of data. Independent ensemble methods are considered more scalable than dependent methods, as the classifier involved in the ensemble approach can be trained in parallel. Usability is another metric that assesses the user’s preference for comprehending how to adjust the ensemble models they employ. Broadly speaking, a good ensemble method should contain a comprehensive set of control parameters that can be easily adjusted.

6. Application domains

This section highlights applications of ensemble learning across different domains, using either traditional or deep learning as baseline classifiers. In general, we briefly summarize the baseline classifiers applied, the ensemble techniques used, and the domain used in their experiments.

6.1. Applications of traditional ensemble learning

This part discusses applications of traditional ensemble learning in various domains, including image classification, natural language processing (NLP), and others. Table 2 summarizes some works that presented ensemble learning methods in machine
Table 2
Applications of ensemble learning in machine learning approach.

<table>
<thead>
<tr>
<th>Studies</th>
<th>Baseline Classifiers</th>
<th>Fusion Method</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shipp and Kuncheva (2002)</td>
<td>NB</td>
<td>Voting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Stamatatos and Widmer (2002)</td>
<td>SVM</td>
<td>Voting</td>
<td>Music Recognition</td>
</tr>
<tr>
<td>Cho and Won (2003)</td>
<td>SVM, KNN</td>
<td>Voting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Wilson et al. (2006)</td>
<td>DT</td>
<td>Boosting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Tsutsumi et al. (2007)</td>
<td>SVM, ME</td>
<td>Stacking</td>
<td>Arabic Sentiment</td>
</tr>
<tr>
<td>Abbasi et al. (2008b)</td>
<td>SVM</td>
<td>Boosting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Li et al. (2010)</td>
<td>SVM, LR</td>
<td>Voting</td>
<td>Chinese Sentiment</td>
</tr>
<tr>
<td>Lu and Tsou (2010)</td>
<td>NB, ME, SVM</td>
<td>Stacking</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Xia et al. (2011)</td>
<td>NB, ME, SVM</td>
<td>Stacking</td>
<td>Named Entity Recognition</td>
</tr>
<tr>
<td>Eckal and Saha (2011)</td>
<td>SVM, NB, ME</td>
<td>Voting</td>
<td>Chinese Sentiment</td>
</tr>
<tr>
<td>Li et al. (2012)</td>
<td>SVM, KNN</td>
<td>Stacking</td>
<td>Chinese Sentiment</td>
</tr>
<tr>
<td>Su et al. (2012)</td>
<td>ME, SVM</td>
<td>Voting, Stacking</td>
<td>Chinese Sentiment</td>
</tr>
<tr>
<td>Hassan et al. (2013)</td>
<td>SVM</td>
<td>Boosting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Rodriguez-Penagos et al. (2013)</td>
<td>SVM</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Clark and Wicentowski (2013)</td>
<td>NB</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Anifowose et al. (2013)</td>
<td>RF</td>
<td>Bagging</td>
<td>Petroleum Reservoir</td>
</tr>
<tr>
<td>Shahzad and Laessens (2013)</td>
<td>NB, DT, KNN</td>
<td>Voting</td>
<td>Malware Detection</td>
</tr>
<tr>
<td>Wang et al. (2013)</td>
<td>SVM</td>
<td>Voting</td>
<td>Image Classification</td>
</tr>
<tr>
<td>Cortes et al. (2014)</td>
<td>DT</td>
<td>AdaBoost</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Kuznetsov et al. (2014)</td>
<td>DT, LR</td>
<td>Voting, Bagging</td>
<td>Medical Sentiment</td>
</tr>
<tr>
<td>Fersini et al. (2014)</td>
<td>SVM, NB</td>
<td>Voting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Wang et al. (2014)</td>
<td>SVM, KNN, DT, ME, NB</td>
<td>Stacking, Boosting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Da Silva et al. (2014)</td>
<td>SVM, RF, LR</td>
<td>Voting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Anwar et al. (2014)</td>
<td>KNN, DT, RF, LR</td>
<td>Voting, Bagging</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Bharathidason and Venkateswaran (2014)</td>
<td>RF</td>
<td>Bagging</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Zareapoor and Shamsolmoalai (2015)</td>
<td>NB, KNN, SVM</td>
<td>Bagging, Boosting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Kanakaraj and Guddeti (2015)</td>
<td>NB, SVM</td>
<td>Bagging, Boosting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Prusa et al. (2015)</td>
<td>KNN, SVM, LR</td>
<td>Bagging, Boosting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Bashir et al. (2015)</td>
<td>SVM, LR</td>
<td>Voting, Bagging</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Bashir et al. (2015)</td>
<td>SVM, DT</td>
<td>Voting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Mishra and Mishra (2015)</td>
<td>NB</td>
<td>Bagging, Boosting</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Kang et al. (2015)</td>
<td>SVM</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Xia et al. (2016)</td>
<td>SVM, LR</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Perikos and Hatzigeloudis (2016)</td>
<td>NB, ME</td>
<td>Bagging</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Fersini et al. (2016)</td>
<td>BLR, NL, RDA, SVM</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Onan et al. (2016)</td>
<td>NB, ME, SVM</td>
<td>Voting, Adaboost, Bagging</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Araque et al. (2017)</td>
<td>SVM</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Dedhia and Ramteke (2017)</td>
<td>SVM, NB, ME</td>
<td>Adaboost</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Oussou et al. (2018)</td>
<td>MNB, SVM, ME</td>
<td>Voting, Stacking</td>
<td>Moroccan Dialect Sentiment</td>
</tr>
<tr>
<td>Saleena et al. (2018)</td>
<td>SVM, RF, NB, LR</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Sharma et al. (2018)</td>
<td>SVM</td>
<td>Bagging</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Foudad et al. (2018)</td>
<td>SVM, NB, LR</td>
<td>Voting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Kulkarni et al. (2018)</td>
<td>SVM, NB, RF</td>
<td>Voting</td>
<td>Text Classification</td>
</tr>
<tr>
<td>Livieris et al. (2019)</td>
<td>KNN, DT</td>
<td>Voting, Bagging</td>
<td>Medical Image</td>
</tr>
<tr>
<td>Chen et al. (2019)</td>
<td>FLDA</td>
<td>Bagging</td>
<td>Groundwater Potential Analysis</td>
</tr>
<tr>
<td>Erdogan and Namli (2019)</td>
<td>SVM</td>
<td>Voting, Stacking</td>
<td>A living environment Analysis</td>
</tr>
<tr>
<td>Seker and Ocak (2019)</td>
<td>RF, LR, Linear R</td>
<td>Bagging</td>
<td>Roadheaders Performance Analysis</td>
</tr>
<tr>
<td>Alrehil and Albalawi (2019)</td>
<td>NB, SVM</td>
<td>Voting, Bagging, Boosting</td>
<td>English Sentiment</td>
</tr>
<tr>
<td>Pasupulety et al. (2019)</td>
<td>SVM, RF</td>
<td>Stacking</td>
<td>India’s Sentiment</td>
</tr>
<tr>
<td>Cai et al. (2020)</td>
<td>SVM, LR</td>
<td>Voting</td>
<td>Chloride Concentration Analysis</td>
</tr>
<tr>
<td>Saeed et al. (2022)</td>
<td>SVM, NB, LR, DT, KNN</td>
<td>Voting, Stacking</td>
<td>Arabic Sentiment</td>
</tr>
</tbody>
</table>

Learning in different fields. In the image classification domain, the researchers in Wang et al. (2013) applied voting based on SVM for image retrieval using COREL images database (Liu et al., 2011). In particular, in medical image classification, the researchers in Cortes et al. (2014) suggested boosting based on deep decision tree (DT) for image classification using several breast cancer datasets. The researchers in Kuznetsov et al. (2014) used AdaBoost based on DT for multi-class classification using 8 UCI datasets (Fernández-Delgado et al., 2014). The researchers in Liu et al. (2015) applied voting and bagging based on kNN and DT to classify lung abnormalities from chest X-rays using three benchmark datasets (Kermany et al., 2018). The researchers in Kermany et al. (2018) proposed bagging based on RF using heart disease dataset (Makhtar et al., 2012). The researchers in Shipp and Kuncheva (2002) proposed voting based on NB using Breast Cancer dataset (Antoniou et al., 2000). The researchers in Mishra and Mishra (2015) applied voting-based NB using six medical image benchmark datasets (Leukemia, Breast cancer, Lung cancer, Hepatitis, Lymphoma, and Embryonal tumors). The researchers in Cho and Won (2003) applied voting based on SVM and KNN using three Leukemia cancer datasets. In Bashir et al. (2015) applied voting and bagging based on SVM and LR using five heart disease datasets. That same year, Bashir et al. (2015) applied voting based on SVM and DT using breast cancer diagnosis datasets. The researchers in Kang et al. (2015) proposed two ensemble methods (bagging and boosting) based on SVMs for the treatment of patients’ diabetes using dataset (Li and Maguire, 2010). In addition, in the NLP domain for the English language, the authors in Wang et al. (2014) used two popular ensemble methods (Bagging, Boosting) based on five base learners (NB, ME, DT, KNN, SVM) by ten public sentiment analysis datasets. The authors in Xia et al. (2011) used stacking based on three algorithms, namely NB,
ME, and SVM, by five datasets. The authors in Li et al. (2010), Xia et al. (2016) applied a voting method based on both LR and SVM using reviews extracted from Amazon.com (Rushdi-Saleh et al., 2011). The authors in Araque et al. (2017) applied voting methods based on different machine classifiers (NB, ME, and SVM) by even public datasets from movie reviews. The authors in Alrehili and Albalawi (2019) suggested three ensemble methods (voting, bagging, and boosting) based on NB and SVM using English customer reviews datasets (Alrehili and Albalawi, 2019). The authors in Saleena et al. (2018) applied voting based on different baseline classifiers (SVM, RF, NB, and LR) by several English tweets datasets. The authors in Dedhia and Ramteke (2017) used AdaBoost based on three classifiers (NB, SVM, and ME) using several English tweets datasets. The authors in Perikos and Hatsizliygeroudi (2016) applied bagging based on NB and ME using different English news portals datasets. The authors in Fersini et al. (2016) used voting based on NB, DT, and SVM by English Movie Reviews datasets (Chen et al., 2012). The authors in Onan et al. (2016) proposed three ensemble methods (bagging, AdaBoost, and stacking) based on five classifiers (BLR, NB, LDA, LR, and SVM) using nine public English sentiment analysis datasets from different domains (Whitehead and Yaeger, 2009). The authors in Kanakaraj and Guddeti (2015) suggested bagging and boosting based on both NB and SVM using English movie review (Pang and Lee, 2005). The authors in Fersini et al. (2014) proposed voting and bagging based on different baseline classifiers (ME, SVM, and NB) by several English movie and product reviews datasets (Täckström and McDonald, 2011; Pang and Lee, 2005). The authors in Prusa et al. (2015) applied KNN, SVM, and LR based on both bagging and boosting using English sentiment140 corpus (Go et al., 2009). The authors in Wilson et al. (2006) introduced boosting based on a DT classifier by English MPQA Corpus (Wiebe et al., 2005). The authors in Tsutsumi et al. (2007) applied stacking based on two classifiers (SVM and ME) using the English movie review dataset (Pang and Lee, 2005). The authors in Hassan et al. (2013) proposed boosting based on SVM using three English product review forum datasets (Abassi et al., 2010; and Abbasi et al., 2008a). The authors in Fouad et al. (2018) compared the performance of a voting method based on three classifiers (SVM, NB, and LR) using several English tweets datasets. The authors in Rodriguez-Penagos et al. (2013) introduced voting based on SVM by English SemEval 2013 dataset (Dzikovska et al., 2013). The authors in Clark and Vicente (2013) suggested voting based on NB using the English SemEval-2013 dataset (Nakov et al., 2016). The authors in Da Silva et al. (2014) applied voting-based four baseline classifiers (SVM, RF, and LR) using several English tweets datasets. But, in multiclass sentiment classification, Sharma et al. (2018) proposed a bagging based on SVM using several English movie review datasets. In contrast, in the Arabic language, the authors in Saeed et al. (2022) applied both voting and stacking for spam detection based on five baseline classifiers (SVM, NB, LR, DT, KNN) using two datasets from Opinion Spam Corpus (Li et al., 2011). Besides, in the different dialects, the authors in Su et al. (2012) applied both voting and stacking based on two algorithms (ME and SVM) using two datasets for three domains of Chinese reviews (book, hotel, and notebook). The authors in Li et al. (2012) suggested stacking based on SVM and KNN using several Chinese food review datasets. The authors in Lu and Tsou (2010) applied stacking based on three classifiers NB, ME, and SVM, using the Chinese dataset (Seki et al., 2008). The authors in Pasupulety et al. (2019) introduced stacking based on two baseline classifiers (SVM and RF) for predicting stock prices of companies using India’s National Stock Exchange (NSE) datasets (Kumar and Misra, 2018). The authors in Oussous et al. (2018) proposed voting and stacking based on three baseline classifiers (MNB, SVM, and ME) using the Moroccan tweets dataset (Tzatz et al., 2015). The authors in Ekbal and Saha (2011) suggested voting based on diverse classification methods such as SVM, ME, and RF for named entity recognition using three Indian languages (Bengali, Hindi, and Telugu) by using Bengali news corpus (Ekbal and Bandyopadhyay, 2008). The authors in Abbasi et al. (2008b) proposed a boosting based on SVM using several middle eastern web forms.

Moreover, in the diverse fields, in Stamatatos and Widmer (2002) used a voting method based on SVM for music performer recognition using several pianists playing datasets. In Chen et al. (2019) applied the bagging method based on Fisher’s linear discriminant function (FLDA) for potential groundwater assessment at the Ningtiaota area in Shaanxi, China. They used using a database with 66 groundwater spring locations. In Zareapoor and Shamsoolmoali (2015) suggested Bagging based on three machine algorithms: SVM, NB, and KNN for credit card fraud predicting. They use 100,000 records of credit card transactions dataset (Hormozi et al., 2013). In Shahzad and Lavesson (2013) proposed voting based on NB, DT, and KNN for malware detection using three datasets of malicious threat (Shahzad et al., 2010). In Afyowose et al. (2013) application bagging RF to predict petroleum reservoir properties using six datasets from a giant carbonate reservoir in the Middle East and a drilling site in the Northern Mar- rion platform of North America (Helmy et al., 2010). In Kulkarni et al. (2018) suggested voting based on SVM, NB, and RF for a crop recommendation system using the input soil dataset into the recommended crop type, Kharif and Rabi. In Ergogan and Namli (2019), applied voting and stacking based on SVM for a living environment prediction. In Cai et al. (2020), voting based on SVM and LR was applied to predict surface chloride concentration. In Seker and Ocal (2019) proposed a bagging based on three classifiers (RF, LR, and Linear R) to predict road headers using several datasets.

6.2. Applications of ensemble deep learning

Ensemble learning methods in deep learning applications outperform traditional ensemble learning in many domains, including image classification, natural language processing (NLP), and others. Table 3 summarizes some works that presented ensemble learning methods in deep learning in different fields. In the image classification domain, in Wang et al. (2020) applied stacking method based on multiple CNNs using CIFAR-10 dataset (Pandit and Kumar, 2020). Also, in Zhang et al. (2019) applied of stacking method based on multiple CNNs used for Image Deburring. They used GoPro dataset (Marques et al., 2021) and the Video Deburring dataset (Wu et al., 2020). In Waltner et al. (2019) proposed boosting method based on CNN used for image retrieval by the biggest available retrieval datasets. In Chen et al. (2019) and Chen et al. (2018) proposed the deep boosting framework by integrating the CNN into the boosting algorithm. They used two benchmark datasets (Set12 and BSD68) (Thakur et al., 2019). In Can Malli et al. (2016) suggested voting based on CNNs for apparent age estimation “face detection” using IMDB-WIKI dataset (Russakovsky et al., 2015). In Opitz et al. (2017) applied Boosting CNNs using several image retrieval datasets (Liu et al., 2016). In Mosca and Magoulas (2016) applied boosting CNN by using two image datasets; namely, MNIST (LeCun, 1998), and CIFAR-10 (Pandit and Kumar, 2020). In Walach and Wolf (2016) proposed boosting CNNs for object counting in images using different image datasets, namely mall crowd counting (Chen et al., 2012). UCf 50 crowd counting (Idrees et al., 2013), UCD (Chan et al., 2005). In Moghimi et al. (2016) applied Boosting CNNs using several image datasets, namely Cars (Krause et al., 2013) and Aircrafts (Gosselin et al., 2014). In Yang et al. (2015) proposed boosting CNNs for face detection using imagenet dataset (Krizhevsky et al., 2012). In Li et al. (2015) suggested stacking based on simi-
Applications of ensemble learning in deep learning approach.

The authors of Ali et al. (2020) applied a boosting method on CIFAR-10 dataset (Pandit and Kumar, 2020) containing 60,000 colored images to train the CNN. In particular, in medical image classification, the authors of Ali et al. (2020) applied a boosting method on the CNN_RNN model using a large document dataset (Dzikovska et al., 2013). In Chen et al. (2017) presented a voting ensemble based on CNN and LSTM using SemEval 2013 dataset (Lewis et al., 2004). In Akhtyamova et al. (2017) suggested voting based on the CNN_RNN model using a large documents dataset (consists of 5871 sentences) for disease prediction related to neuromuscular disorders using deep learning and feature fusion approaches. The proposed system achieved an accuracy of 98.5%. The authors of Alshazly et al. (2019) suggested voting based on CNNs for visual recognition tasks (ear recognition) using several ear datasets.

The authors of Ortiz et al. (2016) applied voting based on deep belief networks using a large dataset from the Alzheimer’s disease Neuroimaging Initiative (ADNI) (Hinrichs et al., 2009). The authors in Xu et al. (2016) presented a voting method based on CNNs for predicting drug safety using deep convex network (DCN) to spoken language understanding (SLU) problems. The stacking method achieved an accuracy of 91.88% by using the ATIS dataset (consists of 5871 sentences). The authors of Wang et al. (2020) suggested a boosting method based on CNNs for disease prediction related to neuromuscular disorders using two neuromuscular disorder datasets (Bakay et al., 2006). In addition, in the NLP domain, in Mohammed and Kora (2021) proposed a novel ensemble for multilingual text classification using six benchmark datasets. Also, compare the performance of the proposed and other ensemble methods. The results prove that the proposed method outperforms the state-of-art ensemble methods. The authors of Alshazly et al. (2019) suggested voting based on CNNs for visual recognition tasks (ear recognition) using several ear datasets.

The authors of Ortiz et al. (2016) applied voting based on deep belief networks using a large dataset from the Alzheimer’s disease Neuroimaging Initiative (ADNI) (Hinrichs et al., 2009). The authors of Codella et al. (2017) proposed voting based on residual networks (DRN) and CNNs for melanoma recognition in dermoscopy images. The voting method achieved an accuracy of 76% by using the dermoscopy images dataset (containing 1279 images) (Mendonca et al., 2015). The authors of Tasci et al. (2021) applied voting based on CNNs for tuberculosis detection by two TB CXR image datasets (Sharma et al., 2017). The voting method achieved an accuracy of 97.5% and 97.68% accuracy rates on datasets, respectively. The authors of Cha et al. (2019) suggested voting based on nine CNNs to classify eardrum and external auditory canal features. The voting achieved an average accuracy of 93.67% by using a large dataset of 910,544 images (Locketz et al., 2016). The authors of Guo et al. (2020) proposed a voting method for automated cervical pre-cancer screening using 30,000 images from several datasets. The voting method combined the assessment of three deep learning architectures, RetinaNet, Deep SVDD, and CNN. The average accuracy and F-score of 91.6% and 0.89%, respectively. The authors of Khamparia et al. (2020) applied a voting method based on CNNs for disease prediction related to neuromuscular disorders using two neuromuscular disorder datasets (Bakay et al., 2006).

In addition, in the NLP domain, in Mohammed and Kora (2021) proposed a novel ensemble for multilingual text classification using six benchmark datasets. Also, compare the performance of the proposed and other ensemble methods. The results prove that the proposed method outperforms the state-of-art ensemble methods. The authors of Mozaffari et al. (2016) applied voting based on CNNs for visual recognition tasks (ear recognition) using several ear datasets.

The authors of Ortiz et al. (2016) applied voting based on deep belief networks using a large dataset from the Alzheimer’s disease Neuroimaging Initiative (ADNI) (Hinrichs et al., 2009). The authors of Codella et al. (2017) proposed voting based on residual networks (DRN) and CNNs for melanoma recognition in dermoscopy images. The voting method achieved an accuracy of 76% by using the dermoscopy images dataset (containing 1279 images) (Mendonca et al., 2015). The authors of Tasci et al. (2021) applied voting based on CNNs for tuberculosis detection by two TB CXR image datasets (Sharma et al., 2017). The voting method achieved an accuracy of 97.5% and 97.68% accuracy rates on datasets, respectively. The authors of Cha et al. (2019) suggested voting based on nine CNNs to classify eardrum and external auditory canal features. The voting achieved an average accuracy of 93.67% by using a large dataset of 910,544 images (Locketz et al., 2016). The authors of Guo et al. (2020) proposed a voting method for automated cervical pre-cancer screening using 30,000 images from several datasets. The voting method combined the assessment of three deep learning architectures, RetinaNet, Deep SVDD, and CNN. The average accuracy and F-score of 91.6% and 0.89%, respectively. The authors of Khamparia et al. (2020) applied a voting method based on CNNs for disease prediction related to neuromuscular disorders using two neuromuscular disorder datasets (Bakay et al., 2006).

In addition, in the NLP domain, in Mohammed and Kora (2021) proposed a novel ensemble for multilingual text classification using six benchmark datasets. Also, compare the performance of the proposed and other ensemble methods. The results prove that the proposed method outperforms the state-of-art ensemble methods. The authors of Mozaffari et al. (2016) applied voting based on CNNs for visual recognition tasks (ear recognition) using several ear datasets.
English reviews from health forums (Karimi et al., 2015). In Araque et al. (2017) applied both voting and stacking based on several deep learning models, namely CNN, LSTM, and GRU, using seven English movie review datasets. In Al-Omari et al. (2019) applied voting based on BiLSTM for English fake news detection using NLP4IF 2019 (Barrón-Cedeno et al., 2019). In Nguyen and Le Nguyen (2019) applied voting based on CNN and LSTM using five English datasets from movie reviews (Koh et al., 2010). In Liervies et al. (2020) proposed CNNs based on bagging and stacking using several English review datasets. In Haralabopoulou et al. (2020) applied both voting and stacking based on several deep learning models, namely LSTM, GRU, CNN, RCNN, and DNN, using two English tweets datasets (SemEval (Bertand et al., 2016), Toxic Comment (van Aken et al., 2018)). In Mohammad and Shaverizade (2021) applied stacking based on four deep learning models, namely CNN, LSTM, GRU, and BiLSTM using English review dataset (SemEval) (Bertand et al., 2016). In Deriu et al. (2017) proposed stacking ensemble based on CNN for English tweets classification by using SemEval-2016 dataset (Bertand et al., 2016). In contrast, in Heikal et al. (2018) applied voting based on the combination of CNN and LSTM models using Arabic dataset (ASTD) (Nabil et al., 2015). In Alharbi et al. (2021) applied a voting method based on LSTM and GRU using five datasets from Arabic tweets.

Moreover, in the diverse fields, in Zhang et al. (2020) proposed a system that jointly learns the grasping and the stacking policies through the grasping for stacking network (GSSNet) for enables a robotic arm to correctly pick boxes from a table and put it on a platform. In Wang et al. (2019) proposed an Adaboost method based on DNN for security level classification. The dataset is the assessment results of 100 Android terminals (including smartphones, smart bracelets, tablet PC) and from schools, hospitals, factories, and other environments. In Liu et al. (2014) applied boosted deep belief network for facial expression recognition/shape changes based on the CK + database (contains 327 expression images) (Seyyedsalehi and Seyyedsalehi, 2014). The authors of Deng and Platt (2014) applied the stacking method based on both RNN and CNN for speech recognition using TIMIT dataset (Garofolo et al., 1993). The authors of Liu et al. (2017) applied stacking based on back propagation neural networks (BPNN) for flood forecasting. Han et al. (2016) applied boosting CNNs for recognizing facial action units. In Tur et al. (2012) applied a stacking method based on deep convex networks (DCNs) to semantic utterance classification by the dataset of utterances from the users of a spoken dialog system. In Palangi et al. (2014) applied stacking RNN for speech recognition systems based on TIMIT dataset (Garofolo et al., 1993).

7. Conclusion

In machine learning, reducing the bias and the variance of models is one of the key factors determining the success of the learning process. In the literature, it has been proven that merging the output of different classification algorithms might decrease the generalization error without increasing the variance of the model. The previous is the key essence of the so-called ensemble learning. Numerous research efforts have preferred ensemble learning over single-model learning in various domains. The main advantage of ensemble learning is combining several individual models to improve prediction performance and obtain a stronger model that outperforms them. In the literature, there are several ensemble techniques to boost classification algorithms. The main difference between any two ensemble methods is training the baseline models and how to combine them. Several research efforts introduced ensemble learning into deep learning models to remedy the problems appearing during the learning process of deep learning models. Usually, the main challenge of deep learning models is that they need a lot of knowledge and experience to tune the optimal hyperparameters aiming at reaching a global minimum error. However, finding the optimal hyperparameters requires an exhausting technique in the search space, which in turn becomes a tedious and time-consuming task. Thus, several research efforts have applied deep ensemble learning in many fields, and most of these efforts are articulated around simple ensemble methods. This paper provided a comprehensive review of the various strategies for ensemble learning, especially in the case of deep learning. This paper also illustrated the recent trends in ensemble learning using quantitative analysis of several research papers. Moreover, the paper offered various factors that influence ensemble methods' success, including sampling the training data, training the baseline models, and the fusion techniques of the baseline models. Also, the papers discussed the pros and cons of each ensemble method. Additionally, the paper extensively introduced and presented several research efforts that used ensemble learning in a wide range of domains and categorized these efforts into either traditional machine or deep learning models as baseline classifiers. It is worth noting that an ensemble of deep learning models using simple averaging methods is not a smart choice and is very sensitive to biased baseline models. On the other hand, Injecting diversity in ensemble deep learning can become robust to the biased baseline models. The diversity can be achieved by training different base-line deep learning architectures over several data samples. The diversity, however, is limited by the computation cost and the availability of suitable data to be sampled.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References


